Location Privacy in Database-driven Cognitive Radio Networks: Attacks and Countermeasures

Zhaoyu Gao†, Haojin Zhu†, Yao Liu‡, Muyuan Li† and Zhenfu Cao†
†Shanghai Jiao Tong University, Shanghai 200240, P. R. China
{gaozy1987, leilmyxwz}@gmail.com, {zhu-hj, zfcao}@sjtu.edu.cn
‡University of South Florida, yliu584@gmail.com

***Abstract*—Cognitive Radio Network (CRN) is regarded as a promising way to address the increasing demand for wireless channel resources. It solves the channel resource shortage problem by allowing a Secondary User (SU) to access the channel of a Primary User (PU) when the channel is not occupied by the PU. The latest FCC's rule in May 2012 enforces *database-driven* CRNs, in which an SU queries a database to obtain spectrum availability information by submitting a location based query. However, one concern about database-driven CRNs is that the queries sent by SUs will inevitably leak the location information.**

In this study, we identify a new kind of attack against location privacy of database-drive CRNs. Instead of directly learning the SUs' locations from their queries, our discovered attacks can infer an SU's location through his used channels. We propose Spectrum Utilization based Location Inferring Algorithm that enables the attacker to geo-locate an SU. To thwart location privacy leaking from query process, we propose a novel Private Spectrum Availability Information Retrieval scheme that utilizes a blind factor to hide the location of the SU. To defend against the discovered attack, we propose a novel prediction based Private Channel Utilization protocol that reduces the possibilities of location privacy leaking by choosing the most stable channels. We implement our attacks and the protection schemes on the data extracted from Google Earth Coverage Maps released by FCC. Experiment results show that the proposed protocols can significantly improve the location privacy.

Keywords – **Location Privacy, database-driven Cognitive Radio Network, Private Information Retrieval**

I. Introduction

Over the last decade, unlicensed channels has been used by prevalent wireless technologies (e.g., Wi-Fi and Bluetooth). However, unlicensed channels only form a small subset of the channel resources that are available to people today. As wireless technologies emerge, unlicensed channels become over crowded. Cognitive Radio Networks (CRNs) thus have been proposed to address the increasing demand for wireless channels and support emerging wireless technologies. CRNs classify two types of users: Primary Users (PUs) and Secondary User (SUs). PUs are licensed users that are pre-assigned with certain channels to operate, and SUs are unlicensed users that are allowed to use PUs' channels only when the channel are not occupied by the PU.

Spectrum sensing and white space database are two typical ways to determine which channels are locally available for reuse by the SUs. In spectrum sensing method, an SU determines whether or not a channel is available by listening to the channel and capturing the PU's signal. In white space database, an SU queries a central database to obtain Spectrum Availability Information (SAI) at his location. The latest FCC's rule [1] in May 2012 eliminates spectrum sensing as a requisite for cognitive radio devices. Instead, it adopts the white space database method and enforces *database-driven* CRNs, in which all fixed or mobile cognitive radio devices are required to query a database to determine available channels. FCC has designated nine entities (e.g. Comsearch, Google Inc.) as TV bands device database administrators. Recently, two TV Bands database systems designed by Koos Technical Services Inc [2] and Telecordia Technologies Inc [3], have been approved by FCC for operation.

Though database-driven CRNs are regarded as a promising approach by following the way of traditional location-based services (LBS), they suffer from privacy threats, especially on the aspect of location privacy. Queries sent by an SU contains his location information. By tracing an SU's database queries, the attacker can geo-locate the SU, and cause other serious privacy leaking if the SU's sensitive data is closely correlated to his location.

Existing approaches for protecting a user's location information in traditional location based services (e.g., k-anonymity approach, or collaborative location privacy protection [4]–[6]) face the challenge of lacking of trusted server or incurring unnecessary cost in a collaborative privacy protection . What's more important, in database driven CRN, as a necessary step for spectrum access, the user should register his using spectrum in the database. In this study, we will show a new kind of location privacy attack, Spectrum Utilization based Location Inferring (SULI) attack, which allows an attacker to infer the location of an SU from the channels he has used. This attack arises from the fact that an SU can gain access to a channel if and only if the presence of the PU is not detected in this location (e.g., out of the coverage of PU). In other words, any event that an SU can or cannot access to a channel with the presence of the PU will leak his location information partially. Such correlation between the spectrum utilization information of an SU and his physical location could be exploited to geo-locate the SU by intersecting the coverage of different channels that the SU has used.

To address the aforementioned challenges, we propose a Privacy Preserving Spectrum Retrieval and Utilization architecture for database-driven CRNs, coined as *PriSpectrum*.

PriSpectrum is comprised of two modules, Private Spectrum Availability Information Retrieval (PSAIR) and Private Channel Utilization (PCU). PSAIR enables an SU to query the database without leaking the SU's location information. The main idea of PSAIR is to utilize a blind factor to hide the location of the SU in the queries. To further mitigate location privacy leaking in spectrum utilization phase, we propose a novel Private Channel Utilization protocol, which defends against SULI attack by always choosing the most stable channel, i.e., the channel with the minimum number of channel switch events. By choosing the most stable channel, PCU dramatically increases the difficulty for the attacker to infer the SU's location. PSAIR and PCU together form a protection layer that can preserve the location privacy of SUs in database-driven CRN.

The contributions of this paper are summarized as below:

1) We identify a new kind of attack against location privacy of database-drive CRNs. Instead of directly learning the SU's location from queries, our discovered attacks can infer an SU's location from the channels that have already been used by the SU.
2) We propose PriSpectrum, a novel protection scheme that can thwart location privacy leaking in database-driven CRNs. PriSpectrum consists of two modules: PSAIR and PCU. The former module deals with privacy leaking from query process, and the latter module combats the new attacks identified in this paper.
3) We perform comprehensive experiments to validate the discovered attack, and evaluate the performance of PriSpectrum. Our experiments are conducted on top of real-world dataset released by FCC on TVFool [7]. The experiment results demonstrate the impact of the discovered attack, as well as the effectiveness and efficiency of PriSpectrum.

The rest of the paper is organized as follows. Section II gives system model and explains our assumptions. Sections III and IV present the new attack and our proposed PriSpectrum protocol, respectively. Section V discusses the experimental evaluation. Section VI concludes this paper.

II. Background and Threat Model

A. Overview of Database Query Process

Database-driven CRNs typically consist of four components: PUs, SUs, Base Station (BS), and Database (*DB*). BS is a radio infrastructure that provides wireless interface and connects SUs' and the database. The BS covered region coined as C is divided into $n \times n$ square cells, each cell is coined as $c_{ij}, i, j \in \{1, 2, \ldots, n\}$, where i is the row index j is the column index. The SAI of the whole BS covered region is stored in the database, which we denote as a $n \times n$ matrix $\mathcal{M}$. The spectrum availability information (SAI) of the cell c_{ij} is coined as m_{ij}. We assume there are K PUs around the BS covered region, which are coined as $PU_k, k \in \{1, \ldots, K\}$, and the channel of PU_k is coined as ch_k. If PU_k is using its channel ch_k, we call that PU_k's state is ON, otherwise, PU_k's state is OFF. The database query process takes three phases: (1) *Query Phase*: an SU sends a query that contains the location c_{ij} of the SU to the BS, who then forwards the query to the *DB*; (2) *Retrieval Phase*: the *DB* retrieves m_{ij}, and sends it back to the SU via BS; (3) *Commitment Phase*: upon receiving the response from the database, the SU chooses an available channel ch_k based on m_{ij} to operate on, and registers the chosen channel ch_k in *DB*. When the state of PU_k changes from OFF to ON, the database will notify the SUs that are using ch_k, and they will launch a new database query process. If ch_k is still available for the SU when PU_k returns, the SU will not change his channel, otherwise he must find another available channel to avoid interference to the PU_k.

B. Threat Model and Assumptions

The attacker's goal is to find the location of a target SU whose position is relatively fixed during a certain interval. Similar to the conventional research on protecting the users' location privacy in LBS, we assume the attacker is a curious whitespace administrator, who collects the locations of customers to make marketing and sales strategies, or external attacker, who harvests the locations of wireless users and sells them for profit. The knowledge of the attackers in this paper are considered to be: 1) the complete communication content between SU and white space DB; or 2) the spectrum utilization information of SUs. Knowing at least one of them will enable the attacker to launch the attack. Note that, an external attacker could obtain spectrum utilization information of a specific SU by simply receiving wireless signals from channels that are being used by SUs and PUs.

We consider a general curious-but-honest model, which means the server will never change the data or query results maliciously. We also assume the attacker has sufficient computational resources such that he can perform real-time analysis and run necessary algorithms to geo-locate SUs.

III. The Discovered Attacks

An SU sends queries that contain its location to the database to retrieve SAI. Thus, an attacker can immediately learn the location of the SU by looking at his queries. Our further investigation into the security of database-driven CRNs discovers more subtle attacks, in which the attacker can obtain the location of an SU without the knowledge of queries. In what follows, we give the overview of the discovered attacks, and present the detailed attack algorithm.

A. Attack Overview

Intuitively, when a PU is ON, and at the same time, an SU could access to the PU's channel, then the SU must be located in the complement of the PU's signal coverage. Otherwise, the SU will cause wireless interference to the PU's transmission. Therefore, by looking at a series of SU's access events, an attacker can narrow down the location range of the SU by intersecting the complements of different PUs' coverage and will eventually get an accurate estimation of the SU's location.

Fig. 1 is a simple attack example that shows the SU's location can be inferred out after the SU accesses four channels. Here, the unavailable area of a channel equals to the coverage of the PU, and the available area is the complement of the PU's coverage. At the beginning, we use 0 to label all the cells in the service area of the Database-driven CRN. At time t_1, the SU accesses to PU_1's channel ch_1. Thus, the attacker can infer that SU is located in cells belong to the complement of the region covered by PU_1. We increase the labels of those possible cells of the SU about ch_1 by 1. At time t_2, the SU accesses to PU_2's channel ch_2. The attacker can then narrow down the location range of the SU to the overlapping cells covered by the intersection of the complement coverage of PU_1 and PU_2. Similarly, we increase the label of the possible cells of the SU about ch_2 by 1. By following these steps for another two accesses to PU_3 and PU_4. Only one cell whose label is largest (i.e. 4) satisfies this condition, thus the SU is located in this cell.

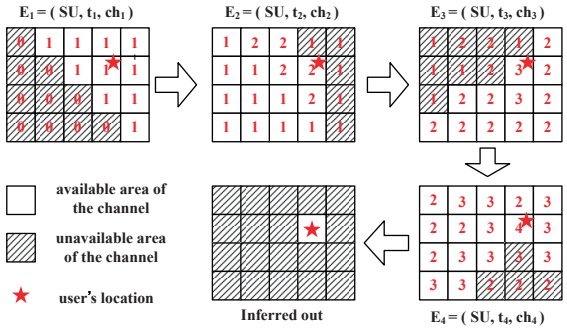


Fig. 1. An example of PUCC attack

.

It should be noted that the attacker can not only use the SU's channel access events, but also use the SU's channel enforced switch events to geo-locate the SU. If an SU switches from one channel to another due to the return of the PU (from OFF to ON), the SU must be located within the coverage of the PU. The attacker may use this observation to further facilitate the localization of the SU.

B. Attack Algorithm

In this section, we propose our *Spectrum Utilization based Location Inferring* (SULI) algorithm that can be used to geo-locate SUs. First we define and formalize the attack approaches into two cases as summarized below.

Primary User Coverage Complement Attack (PUCC): Given PU_k's state is ON, we denote the event that at the time t an SU accesses and uses channel ch_k as *Event I*, which is denoted as $E_t^k = (\text{SU}, t, ch_k)$. Event I indicates that the possible location set S of SU is within the complement area of the coverage of channel PU_k, i.e.,

$$S \in C - C_k \tag{1}$$

,where C and C_k refer to the BS covered region and the coverage of PU_k's signal, respectively.

Algorithm 1: SULI Algorithm

Input: event sequence $E = \{e_1, e_2, \ldots, e_l, \ldots\}$, where e_l could be *Event I* (SU, t, ch_k) or *Event II* $(SU, t, ch_{k_1}, ch_{k_2})$.
Output: the SU's possible location set S.
Initialization: Let SU's possible location set S = C.
Run:
while an event occurs about SU **do**
  if the event is Event I **then**
    PUCC(E_t^k)
  else
    ECS($E_t^{k_1 \to k_2}$)
  end if
end while
function PUCC(E_t^k)
if PU_k's state is ON at time t **then**
  $S \leftarrow S \bigcap (C - C_k)$
end if
end function
function ECS($E_t^{k_1 \to k_2}$)
$S \leftarrow S \bigcap C_k$
PUCC($E_t^{k_2}$)
end function

Enforced Channel Switch Attack (ECS): We denote the event that an SU switches from channel ch_{k_1} to channel ch_{k_2} due to the state transition (from OFF to ON) of PU_{k_1} as *Event II*, which is denoted as $E_t^{k_1 \to k_2} = (\text{SU}, t, ch_{k_1}, ch_{k_2})$. Event II further introduces two situations. If the state of PU_{k_2} is ON, it can be inferred that the possible location set S of SU is within the intersection of PU_k's coverage and the complement of PU_i's coverage, i.e.,

$$S \in C_{k_1} \bigcap (C - C_{k_2}) \tag{2}$$

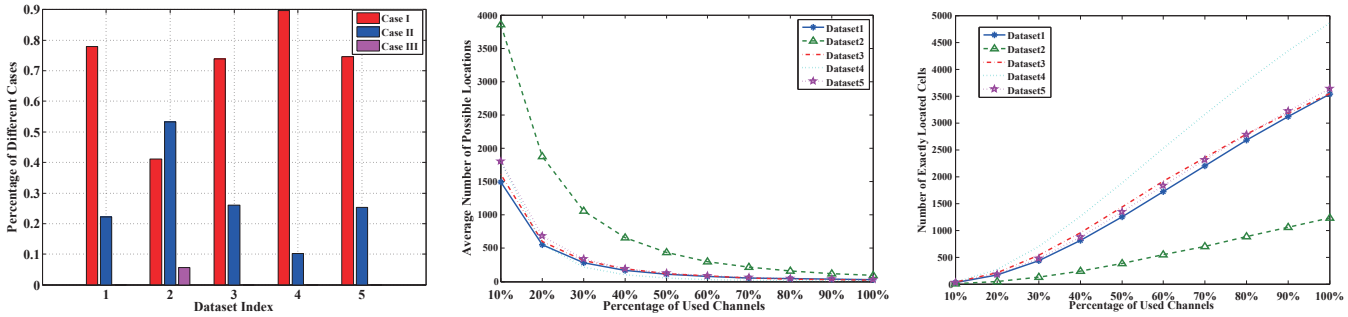
Otherwise, the SU is within the coverage of PU_k, i.e.,

$$S \in C_{k_1} \tag{3}$$

Based on the formalized models of attacks, we give Algorithm 1. Let SU's possible location set S start from C. Thereafter, S shrinks according to the two cases discussed above.

C. Experimental Evaluations

We setup the white space database by adopting the spectrum availability information of Los Angles released on TVFool [7], and implement all FCC restrictions on all TV towers. In LA area, there are 129 channels totally, one of which is shown in Fig.2. Then we extract the SAI from these data and choose 5 sample regions of the scale of $75km \times 75km$. The BS covered region is divided into 100×100 cells, and we set the side length of each cell as 750m which is determined both by the shadowing correlation [8] and the efficiency of spectrum utilization [9]. We perform 20 Monte Carlo experiments by randomly choosing different percentage of channels accessed



(a) Percentage of Each Case (b) Average Result of All Cases (c) Number of Exactly Located Region

Fig. 3. Evaluation Result of the Location Privacy Leakage

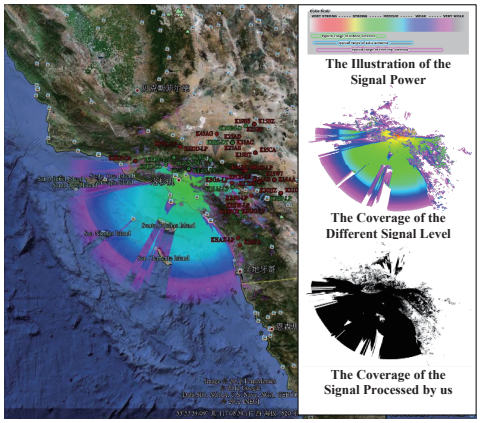


Fig. 2. The coverage of KRCA located at $118.062291°, 34.213338°$ with a 1707.0 m TV tower, whose channel is $ch35$ and ERP is 1000.000 kW

dataset	center location	number of inferred locations	
		average case	Case I
1	$-117.46°, 34.06°$	2.1697	1.7098
2	$-115.82°, 34.06°$	5.3505	2.3107
3	$-117.46°, 32.71°$	2.2292	1.6761
4	$-115.82°, 32.71°$	1.6661	1.5044
5	$-116.78°, 33.39°$	2.1580	1.6279

TABLE I
INFERRED POSSIBLE LOCATION SET WITH ALL THE LEAKED IDENTIFYING CHANNELS

by the secondary users during the presence of PU or enforced channel switch.

We measure the privacy leaking based on spectrum utilization. The results are classified into three categories: the Case I (the good case located to less than 25 cells), the Case II (located to $25 \sim 500$ cells), and the Case III (the bad case located to more than 500 cells).

Table.I shows the inference results in the case that users have traversed all the channels under the presence of PUs. The result shows that the SUs could be located to $1 \sim 2$ cells in Case I while could achieve the localization accuracy of $1 \sim 5$ cells. Fig.3-(a) further gives the proportion of different cases for 5 data sets. It shows that, only one out of total 5 data sets have the Case III, which means, given enough spectrum utilization information, most users could be located with a high accuracy.

We also investigate the situation when no enough spectrum information is provided. We evaluate the average localization performance under the different percentage of channels accessed by SUs. In Fig.3-(b), it implies that, along with the increasing percentage of used channels, the inference accuracy could be improved significantly. Specifically, with more than 50% channel information exploited, SUs could be located to less than 100 cells. Fig.3-(c) shows the number of exactly located SUs (located to only one cell) under different percentage of used channels. It shows that more than 10% regions will be exactly distinguished with only 40% channel information is used.

In our experiments, in 4 data sets, around 80% SUs could be located to less than 10 cells by using 25 or less channels. This further demonstrates the practicality of the discovered attack.

IV. Privacy Preserving Spectrum Retrieval and Utilization for White Space Database

To thwart the previously defined three kinds of location based attacks, in this section, we propose a Privacy Preserving Spectrum Retrieval and Utilization for white space database, which is coined as *PriSpectrum*. As shown in Fig.4, PriSpectrum is comprised of two modules, Private Spectrum Availability Information Retrieval (PSAIR) protocol and Private Channel Utilization (PCU) protocol, which are designed to provide privacy preserving functionality for database query process as well as private spectrum utilization process. Basically, PriSpectrum serves as the frontend between the user and whitespace database to provide privacy preserving function without changing the current whitespace architecture as shown in Fig. 4.

A. Private Spectrum Availability Information Retrieval Protocol

In this section, we first propose a novel PSAIR protocol, which is based on Private Information Retrieval (PIR) technique. The basic idea of PSAIR is to allow an SU to retrieve SAI information of DB without leaking his location information. Then we will discuss the correctness and efficiency of

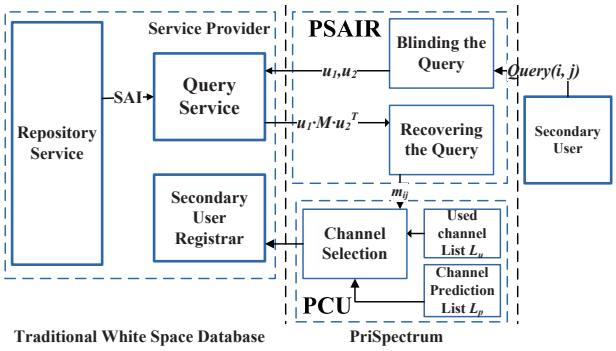


Fig. 4. Overview of PriSpectrum

the proposed scheme, the experiment result of PSAIR will be shown in Section.V.

1) The Process of The PSAIR: Our basic idea is that SU is only interested in the element m_{ij} of the cell c_{ij}, which is the element located at i-th row and j-th column in the $n \times n$ matrix $\mathcal{M}$, but i and j should be kept confidential for DB. In our protocol, an SU sends two vectors without leaking i and j while the SU still could retrieve the SAI m_{ij} he requires. To prevent the DB from guessing i and j the SU is interested in, the SU blinds each element of two vectors with two different blinding factors, which can later be removed by SUs. The detailed protocol is described as follows:

- **System Initialization**: SU chooses a big prime number p, then generates two random numbers b and d as the blinding factors, where $b, d \in \mathbb{Z}_p$, then calculates the inverse of b and d in $\mathbb{Z}_p$ as b^{-1} and d^{-1}. After that, SU generates two n-dimensional random vectors $\vec{v_1} = (a_1, a_2, \ldots, a_n)$ and $\vec{v_2} = (c_1, c_2, \ldots, c_n)$. Here, $a_k, c_k < \frac{\sqrt{p}-\sqrt{N-1}}{nN\sqrt{N-1}}$, where n is the dimension number of the vector or the dimension number of the SAI matrix $\mathcal{M}$, and $N = 2^K$, K is the number of the channels. Actually, N is also the upper bound of the element m_{ij} in $\mathcal{M}$, since each $m_{ij} \in \mathbb{Z}_N$.
- **Query Blinding Phase:** Given SU's location c_{ij}, SU needs to retrieve SAI m_{ij}. He processes $\vec{v_1}$ and $\vec{v_2}$ as

$$\begin{aligned} \vec{v_1}' &= (a'_1, \ldots, a'_n) = N \cdot \vec{v_1} + \vec{h_i} \\ &= (a_1 N, \ldots, a_i N + 1, \ldots, a_n N) \\ \vec{v_2}' &= (c'_1, \ldots, c'_n) = N \cdot \vec{v_2} + \vec{h_j} \\ &= (c_1 N, \ldots, c_j N + 1, \ldots, c_n N) \end{aligned}$$

,where $\vec{h_i}$ is a unit vector with the ith element is 1 and others are 0. To hide the real value of $\vec{v_1}$ and $\vec{v_2}$, SU blinds them by using two blind factors

$$\begin{aligned} \vec{u_1} &= b \cdot \vec{v_1}' \ mod \ p \\ &= (ba_1 N, \ldots, b(a_i N + 1), \ldots, ba_n N) \ mod \ p \\ \vec{u_2} &= d \cdot \vec{v_2}' \ mod \ p \\ &= (dc_1 N, \ldots, d(c_j N + 1), \ldots, dc_n N) \ mod \ p \end{aligned}$$

Then, SU sends the blinded query $Q = (\vec{u_1}, \vec{u_2}, t)$ to *DB*.
- **Query Execution Phase:** After *DB* gets Q, it computes these two blinding vectors with the SAI matrix $\mathcal{M}$, and gets the query result $g = \vec{u_1} \cdot \mathcal{M} \cdot \vec{u_2}^T$ without module operation, where $\vec{u_2}^T$ is the transposition of vector $\vec{u_2}$ Then *DB* sends g back to SU. We will show that this operation could save the transmission cost a lot compared with the direct download $\mathcal{M}$.
- **Result Recover Phase:** To recover the SAI m_{ij}, SU needs to remove the blind fact by multiplying g with b^{-1} and d^{-1}, $g_1 = b^{-1} \cdot g \cdot d^{-1}$. SU modules g_1 with p and N as $m_{ij} = (g_1 \mod p) \mod N$.

Finally SU will choose an available channel ch_k based on m_{ij} and the protocol we proposed in Section.IV.B.

2) Security and Efficiency Discussions: The general security of PSAIR follows the scheme proposed in [10], which is based on the Hidden Modular Group Order assumption. Since we extend the scheme of [10] from single dimension to two dimensions, we then prove its correctness by the following theorem.

Theorem I *Given $\vec{u_1}$ and $\vec{u_2}$ generated by following PSAIR protocol proposed in Section IV. A, SU can retrieve the SAI information on cell m_{ij} correctly.*

Proof: For ease of presentation, we denote $n \times n$ matrix $\mathcal{M}$ by the combination of the row vectors or column vectors as

$$\mathcal{M} = \begin{bmatrix} \vec{r_1} \\ \vec{r_2} \\ \vdots \\ \vec{r_n} \end{bmatrix} = [\vec{e_1}^T, \vec{e_2}^T, \cdots, \vec{e_n}^T]$$

where $\vec{r_i}$ and $\vec{e_i}$, $i \in \{1, 2, \ldots, n\}$ are all row vectors, and $\vec{e_i}^T$ is the transposition of the row vector $\vec{e_i}$.

By following PSAIR protocol, the two blinding query vectors generated by SU are $\vec{u_1}$ and $\vec{u_2}$, which could be denoted as

$$\begin{aligned} \vec{u_1} &= b \cdot (Na_1, Na_2, \ldots, Na_i + 1, \ldots, Na_n) \\ &= bN(a_1, \ldots, a_n) + b(0, \ldots, 0, 1, 0, \ldots, 0) = bN\vec{a} + b\vec{h_i} \end{aligned}$$

$$\begin{aligned} \vec{u_2} &= \cdot(c_1 N, c_2 N, \ldots, c_j N + 1, \ldots, c_n N)d \\ &= (c_1, \ldots, c_n)Nd + (0, \ldots, 0, 1, 0, \ldots, 0)d = \vec{c}Nd + \vec{h_j}d \end{aligned}$$

where $\vec{h_i} = (0, \ldots, 0, 1, 0, \ldots, 0)$ is a unit vector whose i_{th} element is 1. All the operations of $\vec{u_1}$ and $\vec{u_2}$ are in the field of $\mathbb{Z}_p$. Thus the detailed expression of big number operation executed by DB is

$$\begin{aligned} &\vec{u_1} \cdot \mathcal{M} \cdot \vec{u_2}^T \\ &= (bN\vec{a} + b\vec{h_i}) \cdot [\vec{e_1}^T, \vec{e_2}^T, \cdots, \vec{e_n}^T] \cdot (\vec{c}^T Nd + \vec{h_j}^T d) \\ &= (bN[\vec{a}\vec{e_1}^T, \cdots, \vec{a}\vec{e_n}^T] + b\vec{r_i}) \cdot (\vec{c}^T Nd + \vec{h_j}^T d) \\ &= bN[\vec{a}\vec{e_1}^T, \cdots, \vec{a}\vec{e_n}]\vec{c}^T Nd + b\vec{r_i}\vec{c}^T Nd + bN\vec{a}\vec{e_j}^T d + bm_{ij}b \\ &= b \cdot (N[\vec{a}\vec{e_1}^T, \cdots, \vec{a}\vec{e_n}^T]\vec{c}^T N + \vec{r_i}\vec{c}^T N + N\vec{a}\vec{e_j}^T + m_{ij}) \cdot d \end{aligned} \quad (4)$$

In the follows, we denote $(N[\vec{a}\vec{e_1}^T, \cdots, \vec{a}\vec{e_n}]\vec{c}^T N + \vec{r_i}\vec{c}^T N + N\vec{a}\vec{e_j}^T + \mathcal{M}_{ij})$ as m'_{ij}. Since each a_{ij} and c_{ij} must be less than $\frac{\sqrt{p}-\sqrt{N-1}}{nN(N-1)}$, without loss of generality, we denote the biggest element among a_{ij} and c_{ij} as x, we could get

$x < \frac{\sqrt{p}-\sqrt{N-1}}{nN\sqrt{N-1}}$. Furthermore, m_{ij} is an element in $\mathcal{M}$, thus $m_{ij} \leq N-1$. So m'_{ij} could be bounded as

$$m'_{ij} < N - 1 + 2nN(N-1)x + n^2N^2(N-1)x^2 < p \quad (5)$$

Since m'_{ij} is in the modular field $\mathbb{Z}_p$, the user could recover m'_{ij} correctly by query execution. Then we could get m_{ij} by $m'_{ij} \mod N$ correctly. ■

Then, we evaluate the transmission cost of proposed scheme by following theorem.

Theorem II *Given n as the dimension of the matrix $\mathcal{M}$ and p as the big prime chosen by the user, the transmission cost of PSAIR (including uploading and downloading transmission) is bounded by $(2n+3)\lceil \log p \rceil$ bits.*

Proof: In the blind query phase, SU will send two blinding vectors $\vec{u_1}$ and $\vec{u_2}$ to DB. Since the element in the blinding vectors $\in \mathbb{Z}_p$ and the prime p is $\log p$ bit, thus the size of two blinding vectors is less than $2n\lceil \log p \rceil$.

On the other hand, according to the proof of Theorem I, DB will send back a big integer $bm'_{ij}d$ to SU. Since $b, m'_{ij}, d \in \mathbb{Z}_p$, the size of the big prime is less than $3\lceil \log p \rceil$.

So, the total transmission cost incurred by PSAIR is $(2n+3)\log p$ bits. ■

The above theorem gives the transmission cost of the P-SAIR protocol. If compared with k-anonymity based solutions (e.g., query SAI information of k cells in a query) or direct cache [11](e.g., downloading all the database to avoid future query), PSAIR provided a bounded downstream transmission overhead, which is much less than the whole database.

B. Private Channel Utilization (PCU) Protocol

In the previous section, we have presented PSAIR to achieve private channel information retrieval. PSAIR only assures the privacy of SAI query process. However, we have discussed in Section III that the attacker could exploit the channel utilization information to infer an SU's location. In this section, we will discuss how to prevent location privacy leaking and thwart PUCC and ECS attack.

We propose Private Channel Utilization (PCU) protocol, aiming at reducing the location privacy leaking during the spectrum utilization process without changing the existing CR access policy. PCU is motivated from the following observations: 1. The location privacy of SUs will be leaked out only if he accesses to a new channel, which means re-accessing to used channel will not incur new privacy leaking. 2. The expected duration for each channel is naturally diversified. This means that, given a fixed time interval, SUs who choose a more stable channel will have less enforced channel transitions, and thus have less privacy leaking. Based on this observation, we could obtain two principles for spectrum utilization. In particular, to reduce the location privacy leaking in database driven CRN, SUs should choose the channels by following the two principles below

- *Used Channel First:* The channel that has been accessed before is prior to the one that is not accessed.
- *Stable Channel First:* The stable channel is prior to an unstable channel.

With these two principles, we propose PCU algorithm as:

1) The Proposed PCU Algorithm: To follow the two principle, an SU will initialize two lists: a used channel list L_u, which records all of the channels SU has accessed before, and a prediction list L_p, which records all the channels' *predicted duration*. Here, predicted duration is the factor that is introduced to reflect *Stable Channel First* principle. Basically, predicated duration represents how long this channel will be available in the future. We will discuss the details of calculating expected duration in the next section.

Whenever an SU accesses the channel or make a channel switch, SU will firstly try to find the most stable channel from used channel list L_u. If no available channel in L_u, SU will find the most expected stable channel ch_k in L_p. Further, SU will update used channel list by including ch_k in L_u. For each query, SU will update the prediction information in L_p by considering the latest SAI query information. We summarize the PCU algorithm in Algorithm 2.

Algorithm 2: The Proposed PCU Algorithm

Initialization: Let the used channel list L_u be $\emptyset$ and initialize a prediction list L_p through a learning process.
Run:
while input a query result m_{ij} **do**
  if there is an available channel $\in L_u$ **then**
    choose the most stable channel $ch_k \in L_u$.
  else
    choose the most stable channel $ch_k \in L_p$
    put ch_k into L_u.
  end if
  Update L_p by learning.
end while

2) Leveraging Channel Prediction to Choose Most Stable Channel: The remaining problem is how to choose most stable channel from a channel candidate pool. An SU has no idea about the state of the PUs. In stead, it only knows whether the state of each channel ch_k in his cell is available or not. Thus, SAI information of a channel ch_k will be the observation of an SU about PU_k. We model channel ch_k's state as a continuous Markov process with the state transition rate Q-matrix Q_k, which could represented as follows,

$$Q_k = \begin{bmatrix} -\lambda_k & \lambda_k \\ \mu_k & -\mu_k \end{bmatrix} \quad (6)$$

, where λ_k denotes the transition rate from unavailable(0) to available(1) and μ_k denote the transition rate from available(1) to unavailable(0). Then, we obtain the state transition matrix

$$P^k(t) = e^{Qt} = \begin{bmatrix} p^k_{00}(t) & p^k_{01}(t) \\ p^k_{10}(t) & p^k_{11}(t) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\mu_k}{\lambda_k+\mu_k} + \frac{\lambda_k}{\lambda_k+\mu_k}e^{-(\lambda_k+\mu_k)t}, & \frac{\lambda_k}{\lambda_k+\mu_k} - \frac{\lambda_k}{\lambda_k+\mu_k}e^{-(\lambda_k+\mu_k)t} \\ \frac{\mu_k}{\lambda_k+\mu_k} - \frac{\mu_k}{\lambda_k+\mu_k}e^{-(\lambda_k+\mu_k)t}, & \frac{\lambda_k}{\lambda_k+\mu_k} + \frac{\mu_k}{\lambda_k+\mu_k}e^{-(\lambda_k+\mu_k)t} \end{bmatrix}$$

As discuss above, the probability of a channel ch_k changed from available to available is p_{11}^k. We could obtain the expectation of a channel's available duration as:

$$E[t_1] = tp_{11} + 2tp_{11}^2 + \cdots = \frac{tp_{11}}{(1-p_{11})^2} \tag{7}$$

This equation shows that, when an SU needs to choose a channel in a candidate pool, he needs to choose a channel ch_k with the largest probability p_{11}^k to maximize $E[t_1]$.

The next question is how to calculate probability p_{11}^k in our case. If the duration of channel availability and unavailability follows exponential distributions [12], we could estimate λ_k and μ_k by maximum likelihood estimation as follows

$$\lambda_k = \frac{1}{N}\sum_{1}^{N}\sigma(a_{ij}^k = 1) \tag{8}$$

$$\mu_k = \frac{1}{N}\sum_{1}^{N}\sigma(a_{ij}^k = 0) \tag{9}$$

Here, a_{ij}^k refers to the availability information of ch_k in cell c_{ij}, N is the total number of queries, $\sigma(\cdot)$ is a function that equals 1 when function condition is satisfied, otherwise equals 0. According to (8), we could update μ in each query, and thus we could choose a channel with a maximum probability.

V. Implementation and Evaluation

In this section, we evaluate the effectiveness and efficiency of the proposed PriSpectrum from following aspects: 1) Implementation and the performance of the proposed PriSpectrum; 2) Effectiveness of PriSpectrum.

A. Implementation of PSAIR

We implement the proposed PSAIR protocol to achieve Private Spectrum Query Module on OpenSSL C++ big integer library. The evaluation is performed on both of computer and mobile device. The implementation platform includes a 64-bit computer with Intel i5 CPU of 2.8 GHz and 4G memory and an android smart phone with a Qualcomm MSM7201A 528MHz CPU and 192MB RAM,512MB ROM. We evaluate the scalability of the PSAIR protocol under the different parameter setting. In the experiment, we set big prime p as 2048 bits and evaluate the efficiency of three phases under different number of cells and channels as shown in Fig.5.

1) Cost of Blinding Vector Generation on User Side: The first metric is blind vector, which is generated at the user side. We evaluate the cost of blinding vector generation process by evaluating the computation latency. Fig. 5(a) shows the relationship between the blinding vector generation time and the number of cells on a smart phone. It is observed that the computation latency increases from 0.7s to 1.4s when the number of cells increasing from 100^2 to 300^2. It demonstrates a good scalability of PSAIR. Note that, this process could be performed during the offline phase, which could further reduce the computation latency of private spectrum query.

2) Cost of Private Query Execution at Server Side: The second metric is the cost of big integer execution at server side. From Fig. 5(b), it is observed that the computation cost of the server is linear to the number of cells (n^2). It is also noticed that the number of channels that also have a direct impact on the execution cost of server side. When the number of channels are 32 or 64, the computation cost is very close. However, when the number of channels are increased to 128, the computation cost is increased due to the increase of the computation complexity. In general, the cost on the server could be finished in less than 200ms in the experiments.

3) Cost of SAI Recovery at User Side: The last performance metric evaluated is the cost of SAI recovery at user's side, which needs to be performed during the online phase. Fig.5-(c) shows that the execution latency keep relatively stable along with the increase of the number of cells. In particular, SU only needs to spend less than 20 ms even for the largest considered region.

We also evaluate the performance of PSAIR running on PC (Intel CPU i5 of 2.8 GHz) and obtain the results in Table.II. In general, PSAIR achieves a much better performance on PC than the smart phone. The number of channels affects little on the blinding vector generation cost and SAI recovery cost, but it will inevitably increase the cost on the side of server. However, the cost of PSAIR on the server is no more than 120ms, that also demonstrates the efficiency of the PSAIR.

4) Discussions of Transmission Overhead: To demonstrate the transmission overhead, we compare PSAIR with a naive solution, in which SU locally cache the map without any DB query and thus achieve location privacy. In this experiment, it is shown that the SAI size for a given region is more than $40MB$. However, if SU launch a query by following a PSAIR protocol, the overhead could be reduced to 1900 bits, most of which is contributed by cryptographic blind factors. Since it is a bounded transmission cost, the increase of number of cells will not further incur a higher transmission overhead. This further demonstrate the scalability of the proposed solution.

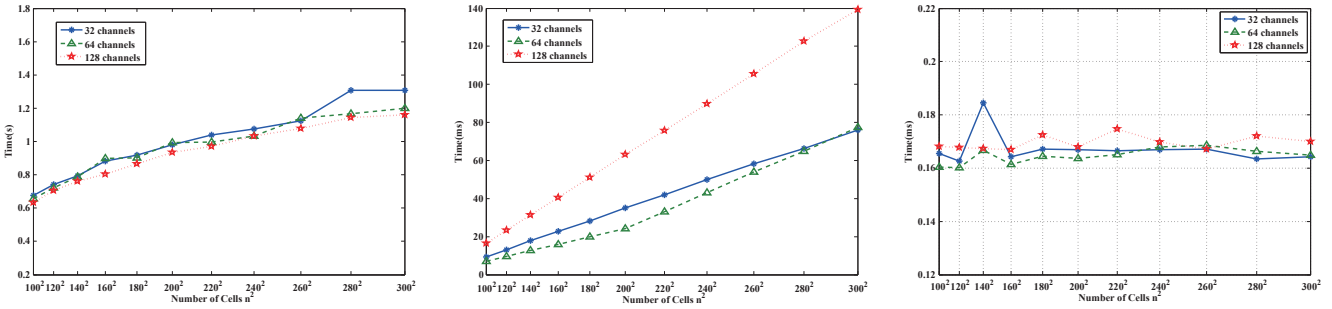
The above discussion demonstrates the efficiency of PSAIR on both sides of SU and server in terms of computation latency. Since the computation cost of PCU is negligible on SU's side, we will evaluate the effectiveness of PCU on protecting SU's location privacy.

B. Evaluate of Private Spectrum Utilization Module (PCU)

We evaluate the private spectrum utilization module by setting up an simulation environment with 10000 SUs uniformly distributed in 10000 cells. Dataset 4 which is faced with the most severe privacy threat is chosen to provide spectrum access information for SUs. The simulation is proceeded for 15 days with the time slot as one minute and the data is collected for every minute. Given the fixed number of SUs in the considered regions, we simulate an area with different user density by tuning the service limitation number, which is originally set to $\tau = 40$ and then increased later. To evaluate the effectiveness of private spectrum utilization module, we implement PCU algorithm with or without Markov

bit number of p	blinding vector generation cost(ms)			server computation cost(ms)			SAI recovery cost(ms)		
	K =32	K=64	K=128	K =32	K=64	K=128	K =32	K=64	K=128
2^{10}	4.505	4.506	4.151	17.517	24.254	38.480	0.056	0.064	0.059
2^{11}	12.907	12.929	12.761	24.030	34.439	62.166	0.163	0.162	0.169
2^{12}	38.700	38.289	38.048	42.012	65.783	117.382	0.563	0.558	0.567

TABLE II
EVALUATION OF PSAIR UNDER THE NUMBER OF CELLS $n^2 = 200^2$ ON PC INTEL CPU I5 OF 2.8 GHZ



(a) Cost of blinding vector smart phone (b) Cost of query execution on server (c) Cost of private SAI recovery on smart phone

Fig. 5. Evaluation of Private Spectrum Query Module

prediction. We compare these two kinds of algorithms with the random channel selection, which randomly chooses a channel to access. The simulation results are shown in Fig. 6.

As introduced in Section IV.B, more channels used will lead to more location information leaking. In Fig. 6-(a), we measure the number of channels have been used before. It is observed that PCU algorithm achieves a significant reduction in terms of number of used channels. Compared with random selection strategy which has used almost 90 channels for 15 days, the number of used channels is bounded to 10 in average. It is also observed that PCU with the Markov prediction incur less used channels than PCU without Markov prediction, which is consistent with our design goal.

The second metric is measurement of location privacy leaking (or geo-location accuracy), which is measured by the size of possible location set. We show the size of possible location set under three channel selection algorithms in Fig. 6-(b). It is observed that the random selection algorithm can enable the attacker to localize a user in less than 100 cells. However, under the protection of PCU algorithms, the attacker can only geo-locate an SU in more than 3000 cells (without prediction) or more than 4000 cells (with prediction).

In Fig. 6-(c), we investigate the number of SUs who are located into less than 25 cells. It shows that in random selection, there are about 5000 users (half of total number) will be under the risk of being located to less than 25 cells after just 4 days. However, if having PCU algorithm, the number of SUs being located to 25 cells is significantly reduced. In Fig. 6-(d), we illustrate an extreme case that the number of SUs are geo-located into a single cell. It is shown that there are 15 percent users that can be located to a cell while almost no user is located by a cell if PCU is in place.

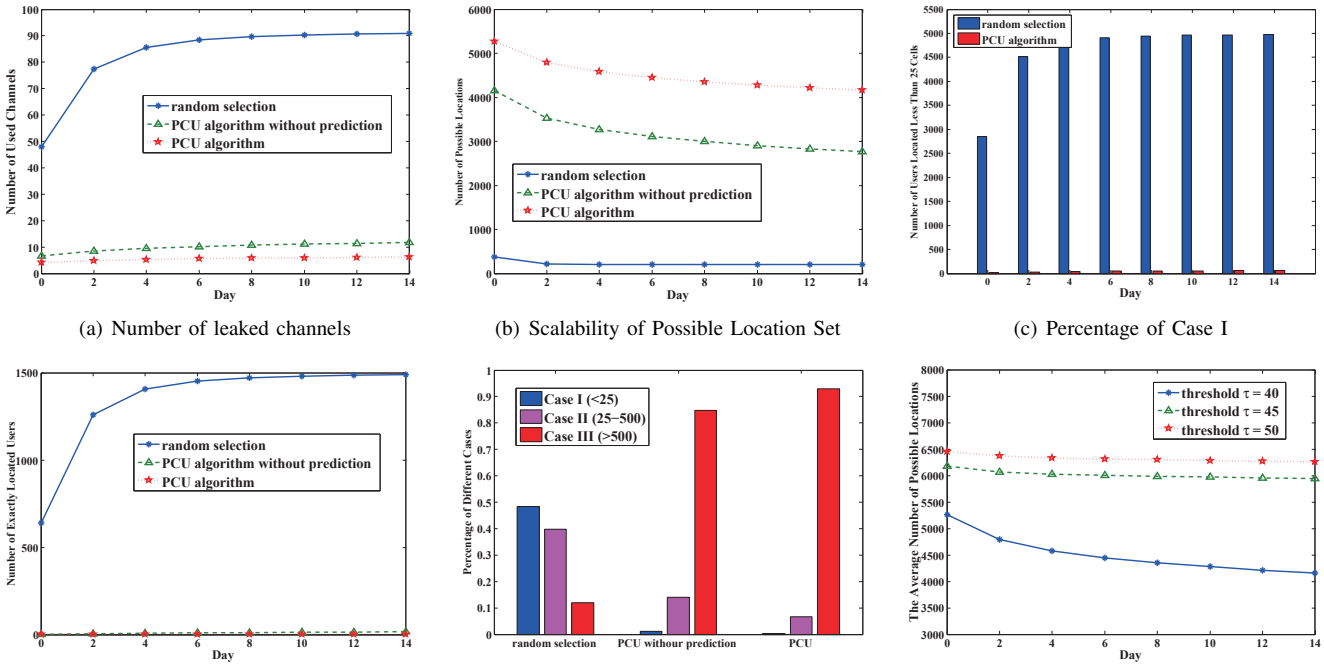
Fig. 6-(e) shows the distribution of inference results with different channel selection algorithm. It shows that, under the attack, most of SUs with random channel selection algorithm could be located to the accuracy of less than 500 cells. However, with PCU, most of SUs are located to larger than 500 cells. This further demonstrates the effectiveness of PCU algorithm. We also investigate the impact of service limitation τ towards the performance of PCU in Fig. 6-(f). It is shown that the smaller the τ is, the better PCU performs. In other words, PCU will achieve a good performance in an area with a low user density, i.e. a rural area.

From simulations, it is shown that the proposed PriSpectrum could well protect the location privacy of SUs with a reasonable cost. It is also observed that PCU with prediction has a better performance than PCU without prediction. This further demonstrates the effectiveness of PCU.

VI. RELATED WORKS

K-anonymity is a widely used privacy protection technique in LBS [4], [5]. K-anonymous location privacy means that the user's location is indistinguishable from at least K-1 other users. To achieve K-anonymous location privacy, one common approach is to incorporate a trusted server, called the anonymizer who is responsible for removing the user's ID and selecting an anonymizing spatial region (ASR) (or cloaking area) containing the user and at least $K-1$ users in the vicinity. However, K-anonymity does not work well in database driven CRNs due to lack of the trusted server.

Collaborative privacy protection such as mix-zone is an alternative approach to k-anonymity in a peer-to-peer fashion [6]. Similar to mixzone based cooperative location privacy in VANET, several SUs could collaborate to change their channels and, thus confuse the mapping of accessed channels before switch and after switch. However, such a collaborative channel switch may not work well in CRNs because it requires



(a) Number of leaked channels (b) Scalability of Possible Location Set (c) Percentage of Case I

(d) Number of Exactly Located SUs (e) The Percentage of Different Cases in Day 4 (f) PCU under Different τ

Fig. 6. Evaluation of PCU

unnecessary channel switches and thus reduce the Quality-of-Service of SUs.

Other research on security issues in database driven CRNs include the latest IETF draft on *protocol to access white space database: security considerations*, which published on July 9, 2012 [13]. In this draft, it discusses the impersonation attacks towards master device, database and man-in-the-middle-attack between SUs and DB. It also suggested using Transportation Layer Security (TLS) protocol to thwart different attacks, i.e. the primary user emulation attack [14]. However, the location privacy concern has not been noticed by [13] in either spectrum sensing of [15] or database-driven CRN.

VII. Conclusion and Future Work

In this paper, we identify a new location privacy attack towards database driven CRNs, which enables the attacker to geo-locate an SU by observing the spectrum he has used. We demonstrate the effectiveness of the discovered attack on spectrum availability information in the area of LA released by FCC. We also propose a novel PriSpectrum scheme to thwart various location privacy attacks. The extensive simulations and experiments well demonstrate the effectiveness and the efficiency of the proposed scheme. Our future work includes other security issues in database-driven CRNs.

References

[1] Federal Communications Commission, "Third Memorandum Opinion and Order," *FCC 12-36*, May, 2012. [Online]. Available: http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-12-36A1.pdf

[2] FCC Encyclopedia, "Chairman Genachowski Announces Approval of First Television White Spaces Database and Device," Dec 22th,2012. [Online]. Available: http://www.fcc.gov/encyclopedia/white-space-database-administration

[3] FCC, "Office of Engineering and Technology Announces the Approval of Telcordia Technologies, Inc.s TV Bands Database System for Operation," March 26th,2012. [Online]. Available: http://www.fcc.gov/encyclopedia/white-space-database-administration

[4] T. Xu and Y. Cai, "Feeling-based location privacy protection for location-based services," in *Proc. of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 348–357.

[5] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for k-anonymous location privacy in participatory sensing," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2399–2407.

[6] J. Freudiger, M. Manshaei, J. Hubaux, and D. Parkes, "On Non-cooperative Location Privacy: A Game-theoretic Analysis," in *Proc. of ACM CCS'09*. ACM, 2009, pp. 324–337.

[7] "TV Fool," March, 2012. [Online]. Available: http://www.tvfool.com/

[8] H. Kim and K. Shin, "In-band Spectrum Sensing in Cognitive Radio Networks: Energy Detection or Feature Detection?" in *Proc. of Mobicom'08*. ACM, 2008, pp. 14–25.

[9] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "Senseless: A Database-driven White Spaces Network," in *Proc. of IEEE DySpan'11*. IEEE, 2011, pp. 10–21.

[10] J. Trostle and A. Parrish, "Efficient Computationally Private Information Retrieval from Anonymity or Trapdoor Groups," *Information Security*, pp. 114–128, 2011.

[11] S. Amini, J. Lindqvist, J. Hong, M. Mou, R. Raheja, J. Lin, N. Sadeh, and E. Tochb, "Cache: caching location-enhanced content to improve user privacy," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, no. 3, pp. 19–21, 2010.

[12] R. Min, D. Qu, Y. Cao, and G. Zhong, "Interference avoidance based on multi-step-ahead prediction for cognitive radio," in *Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on*. IEEE, 2008, pp. 227–231.

[13] Y. C. Y. Wu, "Protocol to Access White Space Database: Security Considerations," July 9, 2012. [Online]. Available: http://datatracker.ietf.org/doc/draft-wu-paws-secutity/

[14] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *Security and Privacy 2010*. Ieee, 2010, pp. 286–301.

[15] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location Privacy Preservation in Collaborative Spectrum Sensing," *Proc. of INFOCOM'12*, 2012.